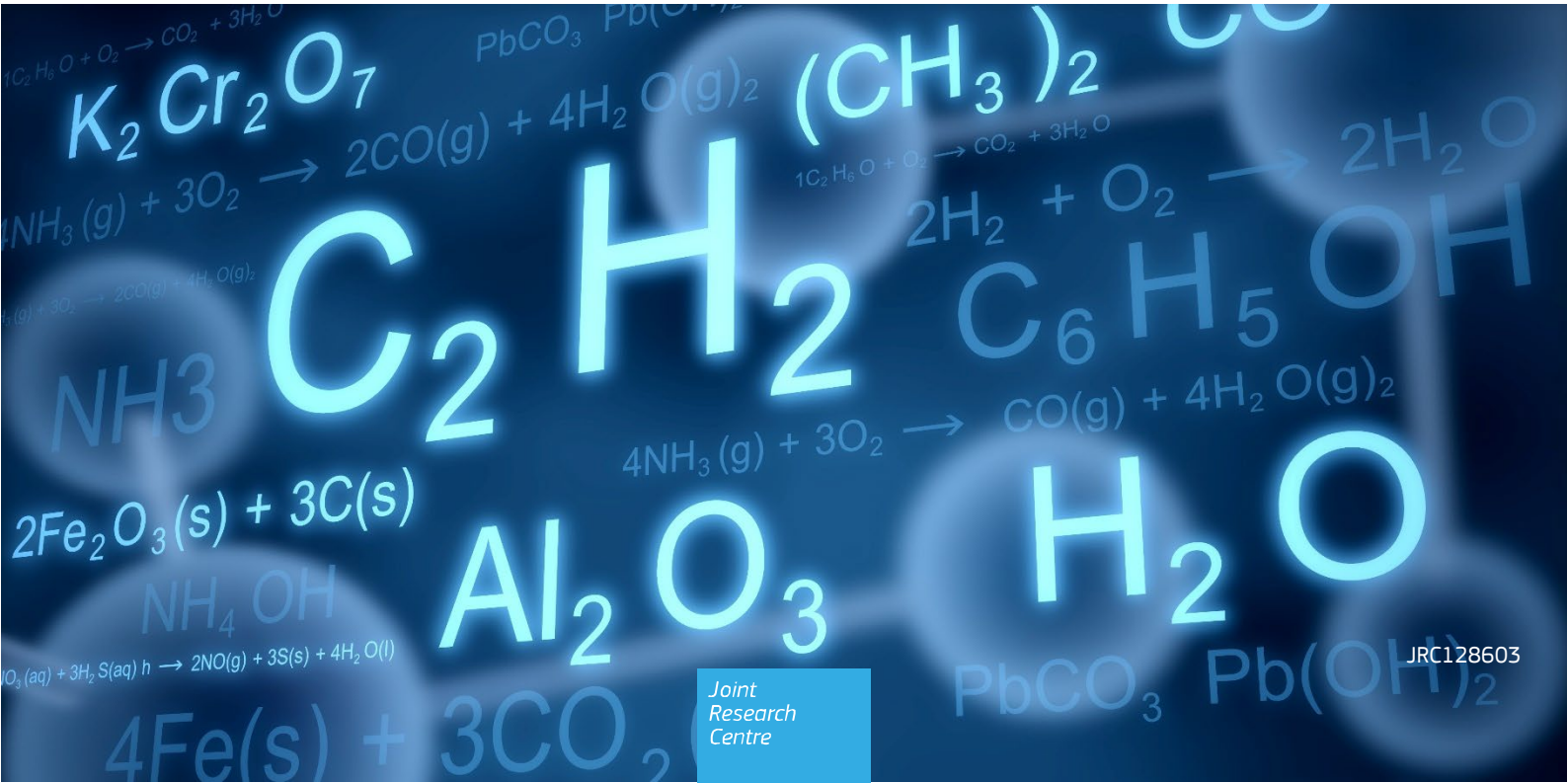
European Commission

JRC SCIENTIFIC INFORMATION SYSTEMS AND DATABASES REPORT

ProofChem: a macro for automated proofreading of chemical formulae

Anderson, D.

2022



Contact information
Name: David Anderson
Address: JRC Geel, Retieseweg 111, Geel, 2440, Belgium
Email: david.anderson@ec.europa.eu
Tel.: +32 14571997

EU Science Hub
https://ec.europa.eu/jrc

JRC128603

Geel: European Commission, 2022

How to cite this report:
Anderson, D., *ProofChem: a macro for automated proofreading of chemical formulae*, European Commission, 2022, JRC128603.

Contents

Acknowledgements

I would like to thank Giovanni Emma for his feedback during the development of this macro and report, and for pointing me (as a non-chemist) in the direction of the recommendations for the notation of chemical formulae published by the International Union of Pure and Applied Chemistry (IUPAC).

Abstract

ProofChem is a productivity macro that aims to automatically detect all chemical formulae in Microsoft Word documents and automatically format each term with the correct subscripts, superscripts, and italics. The use-case scenario is that a document owner runs the macro as a final check that all chemical formulae in a manuscript are correctly formatted, as incorrectly formatted terms can give an unprofessional image. Checking each term manually is a time-consuming and error-prone operation, hence the motivation to automate this process. This macro was developed to help JRC's chemists, physicists, and proofreaders quickly check all chemical formulae (including ions and isotopes) in reports or manuscripts for potential formatting errors. At the time of writing, JRC comprises around 2,800 statutory staff, and produces around 750 technical reports and 900 journal articles per year [JRC, 2022]. Many JRC reports contain dozens, sometimes hundreds of chemical formulae. However, the potential users of this macro are not just limited to JRC researchers. Millions of technical reports and scientific manuscripts containing chemical formulae are published worldwide each year – the majority of which are drafted using MS Word. This report describes how to install and use the macro. A brief, high-level description of the programming principles behind the macro is also provided.

1 Introduction

1.1 What is *ProofChem*?

ProofChem is a productivity macro that aims to automatically detect all chemical formulae in Microsoft (MS) Word documents and automatically format each term with the correct subscripts, superscripts, and italics. The macro acts on body text and text in tables. It does not act on text in textboxes, headers or footers.

1.2 Use case scenario

The use-case scenario is that a document owner runs the macro as a final check that all chemical formulae in a manuscript are correctly formatted, as incorrectly formatted terms can give an unprofessional image. Checking each term manually is a time-consuming and error-prone operation, hence the motivation to automate this process.

1.3 Potential users

This macro was developed to help JRC's chemists, physicists, and proofreaders quickly check all chemical formulae (including ions and isotopes) in reports or manuscripts for potential formatting errors. At the time of writing, JRC comprises around 2,800 statutory staff, and produces around 750 technical reports and 900 journal articles per year [JRC, 2022]. Many JRC reports contain dozens, sometimes hundreds of chemical formulae. For example, in 2021 one JRC Technical Report from the author's unit (JRC.E5 Transport & Border Security) contained over 600 chemical terms with subscripts/superscripts.

However, the potential users of this macro are not just limited to JRC researchers. Millions of technical reports and scientific manuscripts containing chemical formulae are published worldwide each year – the majority of which are drafted using MS Word.

1.4 Notation of chemical formulae

The macro aims to format chemical formulae in line with the following reference document:

Nomenclature of Inorganic Chemistry, IUPAC Recommendations 2005, IUPAC Red Book, prepared for publication by Neil G. Connelly and Ture Damhus (senior editors), and Richard M. Hartshorn and Alan T. Hutton, RSC Publishing, 2005 [ISBN 0 85404 438 8]. https://iupac.org/what-we-do/books/redbook/

The International Union of Pure and Applied Chemistry (IUPAC) is the world authority on chemical nomenclature and terminology, including the naming of new elements in the periodic table; on standardized methods for measurement; and on atomic weights, and many other critically-evaluated data. IUPAC was established in 1919 by academic and industrial chemists who shared a common goal – to unite a fragmented, global chemistry community for the advancement of the chemical sciences via collaboration and the free exchange of scientific information.

Nomenclature of Inorganic Chemistry: IUPAC Recommendations 2005 is the definitive guide for scientists working in academia or industry, for scientific publishers of books, journals and databases, and for organisations requiring internationally approved nomenclature in a legal or regulatory environment. It clarifies and updates recommendations concerning the names and formulae of inorganic compounds and reflects major recent developments in inorganic chemistry. Moreover, it presents recommendations fully consistent with the principles of the nomenclature of organic chemistry.

1.5 Other macro & approaches

There are several other tools in the public domain for automating the formatting of chemical formulae in MS Word, as described below in this sub-section.

One of the simplest is a 20-line macro from April 2019 [Lambert, 2019]. On any selected word, the macro formats all numerals as subscript, and all other characters as normal script. The macro does not discriminate between chemical formulae and other terms, and does not format other terms such as ions.

A more comprehensive macro, called *ChemFormula*, was developed by [Bruner, 2020] & [Pearce, 2018]. The first version of this macro dates from 1998, and the latest version appears to date from July 2012. The macro has to be applied selectively, because it does not check if terms are real chemical formulae. For example, H2020 would be formatted as H_{2020}. When the macro cannot discern if some numerals should be superscript or subscript, and the user is prompted with a dialogue box to decide how to format each term.

Another solution, *Chemistry Formatter*, was distributed as an "Add-in" for MS Word, MS Excel & Powerpoint [King, 2021], and the source code is released under the GPL v. 3 license. The following information is quoted from the macro's website: "*The chemistry formatters are written in the "Visual Basic for Applications" programming language, which is built into Microsoft Office. The code originated from an Excel macro provided by Dr. E. J. Billo, of the chemistry department at Boston College in Chestnut Hill, MA. His macro is also in the CD that comes with his book, Excel for Chemists, 3rd ed, which came out in September 2011.*" The macro seems quite similar to *ProofChem*, although the add-in seems to give incorrect formatting for some ions and isotopes.

The final example of a macro is *ChemFormatter* [Kazuya, 2019] & [ChemFormatter.com, 2019]. The first version appears to be from February 2016, and the latest version appears to be from August 2018. The macro comes with a toolbox of buttons which apply quite sophisticated formatting to selected terms. The main drawback is that the user must select the appropriate button for each term.

It is also worth mentioning a non-macro approach, e.g., [SouthamptonExamFactory. 2021], based on MS Word's in-built AutoCorrect functionality. By adding chemical formulae to the AutoCorrect table, terms will be automatically corrected in real time as one types. The main drawback of this approach is that each term has to be inserted beforehand in the lookup table, and terms that have not been included will not be formatted.

Compared to the approaches described above, the *ProofChem* macro described in this report aims at providing a "one-click solution" to format all chemical formulae in large reports or manuscripts. For this reason, it includes a number of rules (and internal lookup tables) to determine if a term is a chemical formula or not, and a number of rules to determine the formatting of each character.

Finally, it should be emphasised that the notation of chemical formulae can be very complicated and ambiguous, and no automated solution will be error-free. Users of *ProofChem* should review any modifications made by the macro. The author welcomes the reporting of bugs and suggestions for improvements.

2 Installing the *ProofChem* macro

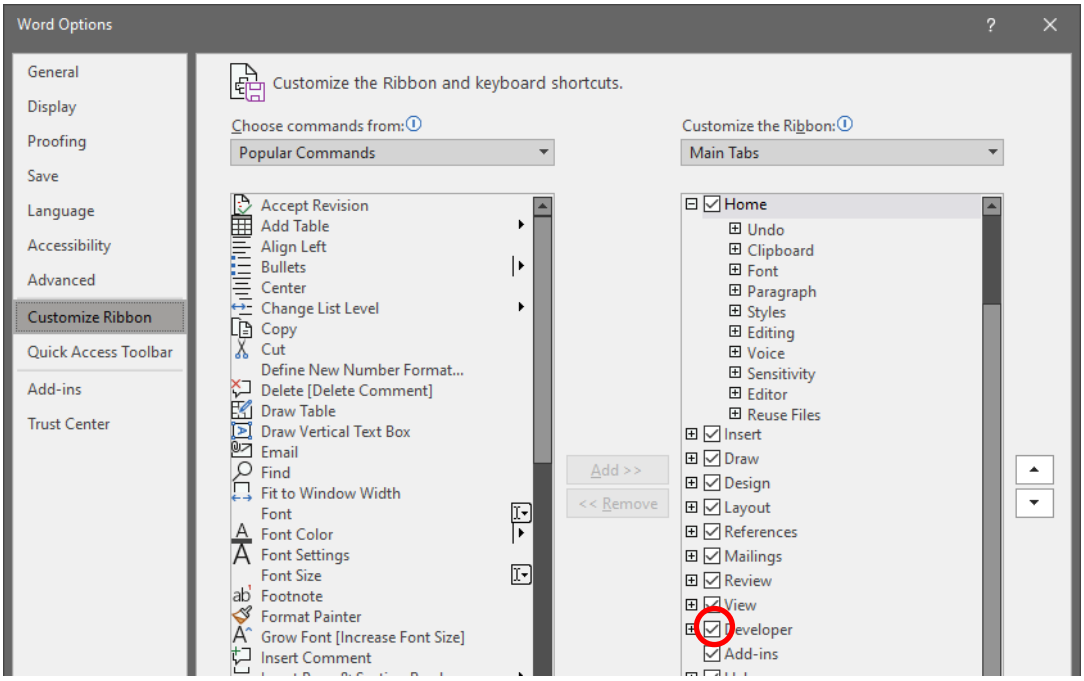
Many corporate IT security settings prevent macros from running directly from a Word file. Often, the code first needs to be copied to the *Normal.dotm* template using the Visual Basic editor. Instructions to do this are given below.

2.1 Installation

Step 1: Display the "Developer" tab in the Ribbon at the top of Word:

— Right click on the ribbon, click Customise the Ribbon and make sure Developer is checked.

Figure 1. How to show the *Developer* tab on the ribbon in MS Word.



Step 2: Copy the macro code

— Open the *.txt file containing the macro code

— Select all text (CTRL+A) and copy it (CTRL+C)

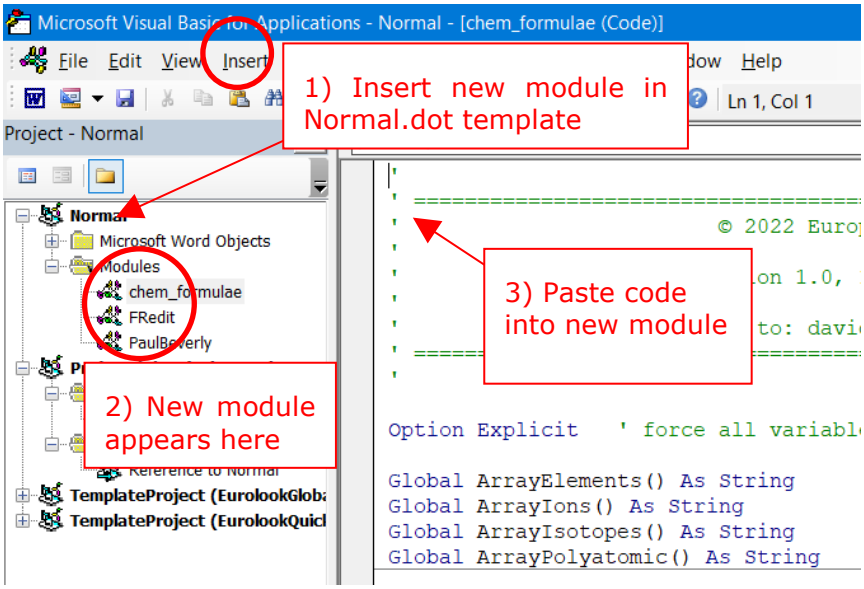
Step 3: Open the Visual Basic Editor view:

— In Word, on the Developer tab, click Visual Basic. Alternatively, press ALT+F11

Step 4: Create a new module and paste the macro code there:

— Select the 'Normal' folder, then click > Insert > Module. The newly created module will have a name like *Module1*. Then paste the code (copied in step 2) into the new module.

Figure 2. How to insert a new module in the Visual Basic Editor to paste the *ProofChem* macro code.



Step 5: Test the macro installation

- Return to Word
- Open any report or manuscript that you want to check
- Launch the macro from the Ribbon:
  - View > Macros > ProofChem, or
  - alternatively, press ALT+F8 and select the macro.

2.2 Creating shortcuts to macros

As for any macro, you can add a macro button to the ribbon to conveniently launch *ProofChem*.

To add a macro button to the ribbon:

1. Click File > Options > Customize Ribbon.
2. Under Choose commands from, click Macros.
3. Click the macro you want.
4. Under Customize the ribbon, click the tab and custom group where you want to add the macro.

3 Using the *ProofChem* macro

3.1 Good practice with symbols

Before running the macro, the user should be aware that inserting symbols in the body text using the *Symbol* font can lead to expected results. Symbols should be inserted using the native font of the text. This is in more detail described below.

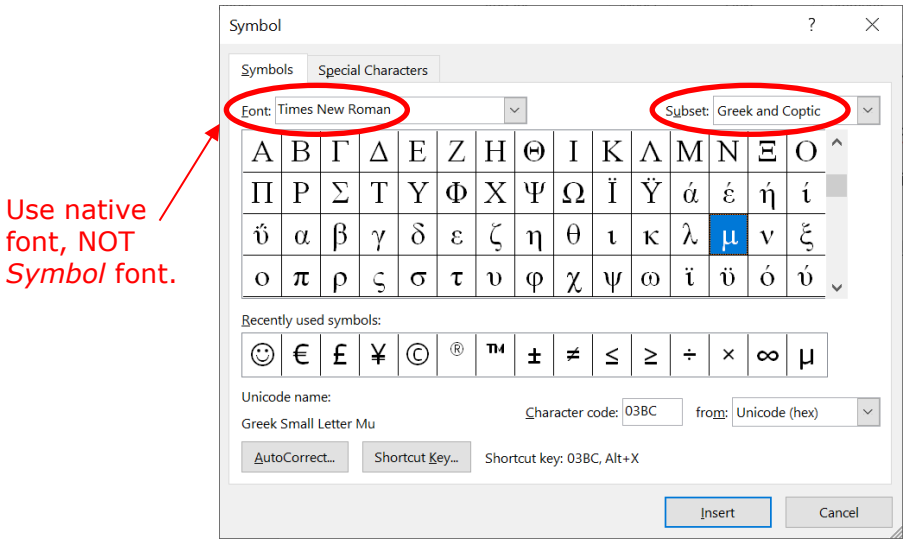
The first task of the macro is to parse the selected text into individual words. A variety of symbols are used as delimiters between words, including spaces, commas, dashes, and also some special characters, such as tabs and line breaks. Furthermore, some symbols play a special role in detecting and correctly formatting chemical formula, such as en dash, em dash, vertically centred dots and µ.

For these reasons, it is important for the good functioning of *ProofChem* to avoid unusual symbols with the *Symbol* font. Before running the macro, symbols should be inserted in the native font (e.g. Arial, New Times Roman, etc.) of the text, and **NOT using the *Symbol* font**, whenever possible.

Example: Inserting Greek letter mu, µ

The correct way to insert, for example, µ, is to call up the symbol dialog box: Insert > Symbol > More Symbols, and then to select the native font (e.g. Arial, New Times Roman, etc.) and **NOT the *Symbol* font**. The Greek letters can be accessed by selected *Greek and Coptic* from the *Subset* dropdown menu (see Figure 3). Alternatively, symbols can be inserted in MS Word using keyboard shortcuts, ALT+*nnnn*, where *nnnn* is the 4-digit ASCII code of the symbol. For example, µ can be inserted by typing ALT+956.

Figure 3. Correct way to insert symbols in MS Word.



Note: The *Symbol* font was established in the early days of word processing, when regular character sets did not include accented letters, Greek letters, and so on. The problem is that characters in *Symbol* font lose their meaning when a new font is applied to the entire paragraph (e.g., α becomes a, β becomes b, etc.). For this reason, character sets were extended to include special characters in the native font. Unfortunately, many people who learned MS Word many years ago still have the habit of inserting symbols using the *Symbol* font.

3.2 What the macro will do

To use *ProofChem*, first select the text you wish to check, then launch the macro. If no text is selected, the macro prompts the user to confirm that the entire document should be analysed.

The macro will scan the document and detect all the instances of chemical formulae, according to an internal system of rules and checks.

Then, each term will be highlighted in yellow, and – where necessary – the characters will be reformatted (subscript, superscript, italics) according to the internal rules. In addition to the yellow highlighting, any term for which one or more characters has been reformatted will be coloured with red font colour.

For example, *ProofChem* will convert the following text:

> *To perform photosynthesis, plants need three things: CO_2, H2O, and sunlight.*

to this:

> *To perform photosynthesis, plants need three things: CO_2, H_2O, and sunlight.*

A full list of the types of chemical formulae that are processed by *ProofChem*, together with 'before and after' examples, is given in Table 1 below. Where appropriate, references are included to the relevant sections of the IUPAC recommendations for nomenclature of inorganic chemistry [IUPAC, 2005].

Table 1. 'Before' and 'after' examples of how *ProofChem* formats chemical formulae. The macro aims at aligning with IUPAC recommendations for nomenclature of inorganic chemistry [IUPAC, 2005].

category / feature	**unformatted (before macro)**	**formatted (after macro)**
normal formulae	H2O AgN3 Pb(N3)2	H_2O AgN_3 $Pb(N_3)_2$
interpunct, i.e. centre dot	H2C2O4·2H2O	$H_2C_2O_4 \cdot 2H_2O$
functional groups	-NO2 –NH–NO2	$-NO_2$ $–NH–NO_2$
empirical formulae	Cs2.5H0.5PW12O40	$Cs_{2.5}H_{0.5}PW_{12}O_{40}$
square brackets & braces [IUPAC 2005, IR-2.2.1]	[SbCl2F] [PtCl2{P(OEt)3}2]	$[SbCl_2F]$ $[PtCl_2\{P(OEt)_3\}_2]$
ligand abbreviations [IUPAC 2005, Table VII]	[Co(en)3]3+ (Ph2PCH2CH2PPh2)2 [CuCl2(NH2Me)2]	$[Co(en)_3]^{3+}$ $(Ph_2PCH_2CH_2PPh_2)_2$ $[CuCl_2(NH_2Me)_2]$

category / feature	**unformatted (before macro)**	**formatted (after macro)**
monatomic ions [IUPAC 2005, IR-4.3]	H+ 2H+ Cr3+ 2Cr3+ Mg2+ Hg22+ 2Hg22+	H^+ $2H^+$ Cr^{3+} $2Cr^{3+}$ Mg^{2+} Hg_2^{2+} $2Hg_2^{2+}$
polyatomic ions	CN- SCN– NO3- CO23- C2H2O2- C22H23Br2N2O3+ Cr2O72-	CN^- SCN^- NO_3^- CO_2^{3-} $C_2H_2O_2^-$ $C_{22}H_{23}Br_2N_2O_3^+$ $Cr_2O_7^{2-}$
coordination complex	[Ag(NH3)]2+ [Ag(S2O3)2]3- Ni[NH2CH2CH2NH2]32+	$[Ag(NH_3)]^{2+}$ $[Ag(S_2O_3)_2]^{3-}$ $Ni[NH_2CH_2CH_2NH_2]_3^{2+}$
isotopes	235U 14C	^{235}U ^{14}C
nuclear isomers	58mCo 234mPa 240fPu	^{58m}Co ^{234m}Pa ^{240f}Pu
isotopically substituted compounds [IUPAC 2005, IR-4.5.2] (Only works for A > 20)	H36Cl 235UF6 32PCl3 K[32PF6] K342K[Fe(14CN)6]	$H^{36}Cl$ $^{235}UF_6$ $^{32}PCl_3$ $K[^{32}PF_6]$ $K_3{}^{42}K[Fe(^{14}CN)_6]$

category / feature	**unformatted (before macro)**	**formatted (after macro)**
isotopically labelled compounds [IUPAC 2005, IR-4.5.3]	H[36Cl] [32P]Cl3 [15N]H2[2H] [13C]O[17O] [32P]O[18F3] Ge[2H2]F2 [36Cl]SOCl2 [2H]PH3 [10B]B2H5Cl	H[^{36}Cl] [^{32}P]Cl_3 [^{15}N]H_2[^{2}H] [^{13}C]O[^{17}O] [^{32}P]O[$^{18}F_3$] Ge[$^{2}H_2$]F_2 [^{36}Cl]$SOCl_2$ [^{2}H]PH_3 [^{10}B]B_2H_5Cl
oxidation states [IUPAC 2005, IR-4.6.1] Note: V and I are also symbols for vanadium and iodine, hence by default are not formatted.	MnVII O–II Ni0 [Os0(CO)5] K[OsVIII(N)O3] PbII2PbIVO4 [Mn-I(CO)5]–	Mn^{VII} O^{-II} Ni^{0} [$Os^{0}(CO)_5$] K[$Os^{VIII}(N)O_3$] $Pb^{II}_2Pb^{IV}O_4$ [$Mn^{-I}(CO)_5$]$^-$
optically active compounds [IUPAC 2005, IR-4.6.3]	(+)589-[Co(en)3]Cl3	$(+)_{589}$-[$Co(en)_3$]Cl_3
excited states [IUPAC 2005, IR-4.6.4]	He* NO*	He^* NO^*
structural descriptors [IUPAC 2005, IR-4.6.5]	cis-[PtCl2(NH3)2] trans-[PtCl4(NH3)2]	*cis*-[$PtCl_2(NH_3)_2$] *trans*-[$PtCl_4(NH_3)_2$]
μ to designate atom or group bridging coordination centres [IUPAC 2005, IR-4.6.5]	[(H3N)5Cr(μ-OH)Cr(NH3)5]5+	[$(H_3N)_5Cr(\mu\text{-}OH)Cr(NH_3)_5$]$^{5+}$
examples of potential false alarms (these should NOT be formatted with subscripts or superscripts)	C4 Hg03 H2020 A+ A* JRC123456 -III, -II, -I, 0, I, II, III, IV, V, VI, VII, VIII,	C4 Hg03 H2020 A+ A* JRC123456 -III, -II, -I, 0, I, II, III, IV, V, VI, VII, VIII,

3.3 Exceptions and limitations

There are two categories of chemical formulae where the notation might be ambiguous and *ProofChem* might give the wrong results.

— Firstly, for **oxidation states** (indicated by Roman numerals), it should be noted that V and I are also the symbols of the elements **vanadium** and **iodine**. Since vanadium and iodine are quite common elements, and the use of Roman numerals to indicate oxidation states is quite uncommon, a design decision was taken to always interpret V and I as element symbols. This default behaviour can be changed in the source code by setting the global variable *VIalwaysElements* to *False*. Alternatively, checking for oxidation states can be disabled completely by setting the global variable *CheckOxidationStates* to *False*.

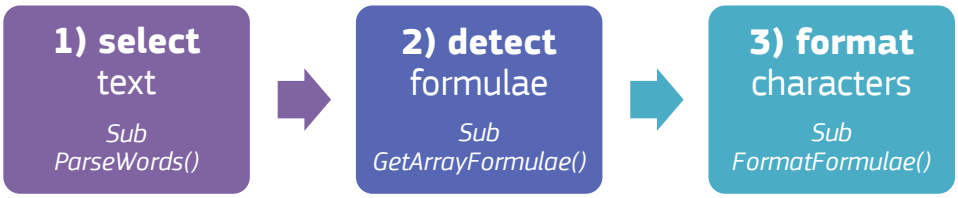
— Secondly, for **isotopically substituted compounds** (e.g. $^{32}PCl_3$), there is the possibility of ambiguous terms when a number precedes an element, or occurs between two elements. The number might refer to the number of molecules in a reaction, the number of the preceding atoms in a molecule, or accompany the subsequent element to indicate an isotope. Compare, for example, ^{2}H, $2H_2O$, and $H^{3}HO$. A design decision was taken only to consider numbers above 20 as possibly indicating an isotopically substituted compound. This default behaviour can be modified in the source code by changing the variable *IsotopeThreshold* = 20 to some other number.

  Note: for specifically labelled compounds, e.g., $[^{2}H]PH_3$, the isotopes are always enclosed in square brackets, and are therefore always recognised and formatted correctly.

4 How does *ProofChem* work?

In this section, a brief, high-level description of the *ProofChem* macro is given. The macro can be split into three main tasks, as illustrated in Figure 4.

Figure 4. Programming flow of *ProofChem* macro.



The first task of the macro is to parse the selected text into individual words. A variety of symbols are used as delimiters between words, including spaces, commas, dashes, and also some special characters, such as tabs and line breaks.

4.1 Detection of chemical formulae

The second task is to determine which words are chemical formulae. For each word in the selected text, a series of tests is performed to identify if the word is a chemical formula. If any test is failed, the word is discarded. If all tests are passed, the word is added to an array of terms that are considered chemical formulae. A description of the tests is given in Table 2.

Table 2. List of 10 tests performed to determine if a word is a chemical formula.

Test no.	Description	Examples	
		pass	fail
1	ignore words that contain characters illegal characters	CO2	CO#2
2	ignore words that don't contain a capital letter	H2O	h2o
3	ignore words that don't contain a numeral (exceptions: ions without numerals, e.g. H^+, F^-, asterisk for excited states, e.g. He*, NO*, Roman numerals for oxidation state, e.g. Mn^{VII}, O^{-II})	H2O	HO
4	ignore words with forbidden sequences, e.g. lowercase letter preceded by a number, or two lowercase letters.	AgN3	Covid-19
5	check that all letters are real chemical elements	H2O2	A2O2
6	ignore words containing numbers with more than three digits	SiO2	H2020
7	ignore words containing numbers if the first digit (of many) is zero	H2SO3	H0123

Test no.	Description	Examples	
		pass	fail
8	if word is <element><number>, check if on the whitelist of polyatomic elements	H2	H9
9	if word is <element><element><number>, discard if number is greater than 8	NH3	NH15
10	if word is <number><element>, check if on the whitelist of isotopes	42K	4K

4.2 Formatting of chemical formulae

The final step is to format each character in each word with the correct superscript, subscript, or italics. This is done by first creating a semantic representation of each word using a word code (*WrdCode*), then translating this to a format code (*FmtCode*).

The word code is created using a function called *GetWrdCode*, which returns a word code where each character is represented with one of the following characters:

- E = uppercase letter (E for Element)
- e = lowercase letter
- N = Numeral
- B = round Brackets
- Q = sQuare brackets
- S = plus or minus Sign (including weird dashes)
- P = decimal Point
- m = lowercase m or f (for nuclear isomers)
- * = asterisk (for excited states)
- g = Greek letter (micron)
- i = geometrical or structural affixes (e.g. *trans*-)
- r = Roman numerals (for oxidations state)
- X = any other character

Example: GetWrdCode("Ag(NH3)2+") would return "EeBEENBNS"

Next, a function called *GetFmtCode* takes a word and its word code to create a format code, where:

- n = normal
- u = superscript (up)
- d = subscript (down)
- i = italic

Example: GetFmtCode("H_2O") would return "ndn"

The function *GetFmtCode* contains a number of rules to decide what kind of chemical formulae a word is (e.g. regular, ion, coordination complex, isotope, etc.), and determine the correct formatting code for each character in the word (e.g. normal, superscript, subscript, etc.). To illustrate the concept, but without going into the full details, some example rules for ions are listed in Table 3 below.

Table 3. Examples showing how word codes are used to determine format codes for various ions. As a general rule, numerals in chemical formulae are subscript, except those cases highlighted in yellow, which are superscript.

type	word code	format code	examples
monoatomic ion	ES	nu	Na^{+}, F^{-}
	ENS	nuu	Mg^{2+}, N^{3-}
polyatomic ion	~EES	~nnu	CN^{-}, OH^{-}
	~EENS	~nndu	NO_3^{-}, ClO_2^{-}
	~EENNS	~nnduu	SO_4^{2-}, PO_3^{3-}
	~NENNS	~dnduu	$Cr_2O_7^{2-}$
complex ion (parentheses)	~BNS	~ndu	$Ag(NH_3)_2^{+}$
	~BNNS	~nduu	$Ag(S_2O_3)_2^{3-}$
coordination complex (square brackets)	~QS	~nu	$[Ag(CN)_2]^{-}$
	~QNS	~nuu	$(Cu(NH_3)_4]^{2+}$
	~QNNS	~nduu	$Ni[NH_2CH_2CH_2NH_2]_3^{2+}$

Note 1: For word codes, E=element, N=numeral, B=round bracket, Q=square bracket, S=sign (plus or minus).
Note 2: For format codes, n=normal, u=superscript, d=subscript.

In addition to the rules-based approach outlined in Table 3, the *ProofChem* macro also makes use of brute-force approaches for other kinds of chemical formulae. For example, an internal lookup table of isotopes is used to determine if a term is an isotope (and subsequently apply the correct superscript, e.g. ^{14}C.)

References

Bruner, B. (2020). *ChemFormula*. Retrieved from http://bbruner.org/download.htm

ChemFormatter.com (2019). Software for Chemistry – *ChemFormatter*. Retrieved from https://www.chemformatter.com/chemformatter

IUPAC. (2005). *Nomenclature of Inorganic Chemistry, IUPAC Recommendations 2005, IUPAC Red Book*. Retrieved from https://iupac.org/what-we-do/books/redbook/

JRC. (2022). *JRC Publications Repository*. Retrieved from https://publications.jrc.ec.europa.eu/repository/

Kazuya, U. (2019). *ChemFormatter download | SourceForge.net*. Retrieved from https://sourceforge.net/projects/chemformatter/

King, C. (2021). *Chemistry Formatter*. Retrieved from https://christopherking.name/ChemFormat/

Lambert, P. (2019). *Macro for Chemical Notation in MS Word « Blog of Peter Lambert.* Retrieved from http://peterl.xyz/2019/04/macro-for-chemical-notation-in-ms-word

Pearce, G. (2018). *ChemFormula*. Retrieved from http://www.gregpearce.co.uk/chemformula/

SouthamptonExamFactory. (2021). *Use AutoCorrect to format formulae as you write*. Retrieved from https://southampton-exam-factory.com/2021/02/15/215/

List of abbreviations and definitions

EU	European Union
IT	information technology
IUPAC	International Union of Pure and Applied Chemistry
JRC	Joint Research Centre
MS	Microsoft

List of figures

List of tables

The European Commission's science and knowledge service

Joint Research Centre

JRC Mission

As the science and knowledge service of the European Commission, the Joint Research Centre's mission is to support EU policies with independent evidence throughout the whole policy cycle.

EU Science Hub

ec.europa.eu/jrc

@EU_ScienceHub

EU Science Hub - Joint Research Centre

EU Science, Research and Innovation

EU Science Hub